ISSN 1009-3095 *Journal of Zhejiang University SCIENCE* V.3, No.4, P.431-434, Sep.-Oct., 2002
http://www.periodicals.com.cn; http://www.zju.edu.cn/English
http://www.zjupress.com; http://lib.zju.edu.cn/eindex.htm; jzu_s@mail.hz.zj.cn

Improved genetic operator for genetic algorithm

LIN Feng (林 峰)†, YANG Qi-wen (杨启文)
(*College of Electrical Engineering, Zhejiang University, Hangzhou 310027, China*)
†E-mail: eeflin@zjuem.zju.edu.cn
Received July 5, 2001; revision accepted Nov. 20, 2001

Abstract: The mutation operator has been seldom improved because researchers hardly suspect its ability to prevent genetic algorithm (GA) from converging prematurely. Due to its importance to GA, the authors of this paper study its influence on the diversity of genes in the same locus, and point out that traditional mutation, to some extent, can result in premature convergence of genes (PCG) in the same locus. The above drawback of the traditional mutation operator causes the loss of critical alleles. Inspired by digital technique, we introduce two kinds of boolean operation into GA to develop a novel mutation operator and discuss its contribution to preventing the loss of critical alleles. The experimental results of function optimization show that the improved mutation operator can effectively prevent premature convergence, and can provide a wide selection range of control parameters for GA.

Key words: Genetic algorithm (GA), Mutation operator, Premature convergence
Document code: A **CLC number:** TP273

INTRODUCTION

Genetic algorithm (GA) is a general methodology for searching a solution space in a manner analogous to the natural selection procedure in biological evolution (Holland, 1975). GA differs from many traditional optimization algorithms which usually suffer from myopia because of highly complex search spaces (Miller et al., 1993). The prominent characteristic of GA is that it tests and manipulates a set of possible solutions simultaneously which assures that GA finds the optimal solution that cannot be found by "hill-climbing" search algorithms or "gradient descent" techniques in some cases (Eshelman et al., 1991).

Premature convergence is a common phenomenon in GA. Grefenstette et al. (1985; 1986) viewed optimization problems with GA as an adaptive system. We prefer to regard the system as a nonlinear system in which GA is a nonlinear controller. This can helpfully explain why different initial conditions and different control parameters of GA can lead to different results.

How can we design a powerful controller to control the optimization process so that the nonlinear system can converge to the optimal solution as soon as possible? Potts et al. (1994) summarized the causes of premature convergence and concluded that the main cause of premature convergence is the loss of critical alleles.

Previous researchers' efforts to prevent premature convergence (Back et al., 1991) included improving

1. selection strategy
2. crossover model
3. probabilities of crossover and mutation

In the usual version, the mutation operator has the ability to exploit the critical alleles. So researchers seldom suspected the ability of the traditional mutational operator to prevent premature convergence and have not tried to improve the traditional mutation operator. In this paper, we analyse the drawbacks of the traditional mutation operator, and present a novel improved mutation operator whose efficiency was verified by some comparison experiments.

DEFECTS OF THE TRADITIONAL MUTATION OPERATOR

The common course of evolution in GA consists of crossover, mutation and selection. As we all know, new chromosomes can be generated after crossover, but no novel genes can be yielded because the goal of the crossover process is just to exchange parts of genes between parents. Selection strategies can bring neither new chromosomes nor new genes into a population. In the stage of selection, GA only selects the higher fitness chromosomes from the contemporary population to reproduce. So GA cannot generate new genes for some loci after reproduction. On the contrary, critical alleles in some loci will disappear with the death of "bad" individuals because of selection. Therefore the exploitation of critical alleles depends on the mutation operator. The traditional mutation operator performs *NOT* operation. This kind of genetic operator, on the one hand, is helpful for finding the critical alleles when premature convergence appears, but on the other hand, it may hurt the critical alleles during their mutation.

For an optimization problem, suppose that the optimal chromosome in GA is S^*, whose length is l.

$$S^*: \quad g_1^* \quad g_2^* \quad \cdots \quad g_i^* \quad \cdots \quad g_l^*$$

The gene $g_i^* \in \{0, 1\}$ denotes the critical gene in the ith locus. For GA, the goal is to find out the critical gene for each locus in the chromosome so that S^* can be built up.

Assumed that population size is N. At the tth generation, the N chromosomes are

$$\begin{array}{lcccccc}
S_1: & g_1 & g_2^* & \cdots & g_i & \cdots & g_l^* \\
S_2: & g_1 & g_2^* & \cdots & g_i & \cdots & g_l^* \\
\vdots & \vdots & & & \vdots & & \vdots \\
S_j: & g_1 & g_2^* & \cdots & g_i^* & \cdots & g_l \\
\vdots & \vdots & & & \vdots & & \vdots \\
S_N: & g_1^* & g_2 & \cdots & g_i^* & \cdots & g_l^*
\end{array}$$

Without loss of generality, we discuss the case of mutating in ith locus for all chromosomes. We hope that the genes in the locus will converge to g_l^*. In other words, all genes of N chromosomes in the same locus become g_i^* after mutation. If the genes are g_i, the invalid genes for the locus, we consider it is premature convergence of gene (*PCG*) in the locus.

Theorem The probability of *PCG* caused by NOT operator is equal to or more than p_m^N, where p_m represents the mutation probability.

Proof Suppose that there are n "0"s and N-n "1"s in ith locus at tth generation. After mutation, the probability that there will be N "0"s (or "1"s) in the locus is determined by:

$$\begin{cases} P(\text{"0"}) = (1-p_m)^n p_m^{N-n} \\ P(\text{"1"}) = p_m^{\ n}(1-p_m)^{N-n} \end{cases} \tag{1}$$

where $P(\bullet)$ denotes the probability of becoming "$\bullet$" in the same locus.

Usually p_m is far less than 0.5 in GA. Thus we have

$$1-p_m \geqslant p_m \tag{2}$$

Substituting (2) into Eq. (1), we can obtain

$$\begin{cases} P(\text{"0"}) \geqslant p_m^N \\ P(\text{"1"}) \geqslant p_m^N \end{cases}$$

Therefore, whatever the invalid gene is, the probability of PCG satisfies

$$P(g_i) \geqslant p_m^N \tag{3}$$

From Eq. (3), we can draw a conclusion that, for the traditional mutation operator, low mutation probability and large population size redound to reduce the probability of PCG. When we adopt the NOT mutation operator, it is not hard to understand why large size of population in GA is preferred, and why the probability of mutation usually ranges from 0.005 to 0.01.

IMPROVED MUTATION OPERATOR

As we all know, invalid genes occupy some loca when GA converges prematurely. From the viewpoint of preventing premature convergence, it is important to maintain the diversity of genes in the same locus rather than the diversity of individuals in the population. Since we cannot identify which kind of genes is critical in a certain locus, we had better enable the alleles to exist in the same locus during the period of mutation.

We present a new mutation operator which is made up of two boolean operators $XOR/\overline{XOR}$ expressed as

$$\begin{array}{ll} XOR: & \begin{cases} a \otimes b = 0, \text{ if } a = b \\ a \otimes b = 1, \text{ if } a \neq b \end{cases} \\ \overline{XOR}: & \begin{cases} a \circ b = 1, \text{ if } a = b \\ a \circ b = 0, \text{ if } a \neq b \end{cases} \end{array} \tag{4}$$

This is a mutation operator different from the traditional one made up of one boolean operator: NOT. Obviously mutation with the new genetic operator needs parents to provide two genes. According to Eq.(4), the result of mutation is that the mutated genes in the same locus of two offspring are in the state of compensation. So provided that there is a pair of genes mutated in the locus, there will be at least one critical allele coming into being in the same locus after mutation. The probability of the loss of critical alleles caused by the improved mutation operator can reduce to zero. As a result, the new mutation operator can to a high degree prevent premature convergence. Table 1 is an example that the genes in the 4th and 7th locus undergo mutation respectively.

Table 1 The genes mutation

Oprerator	Parents	Offspring
XOR	0111010	0110011
$\overline{XOR}$	1101011	1101010

Before mutation, there are two different genes in the 4th locus and genes in the 7th locus are the same while they are mutually exclusive in their own locus after mutation.

EXPERIMENTAL RESULTS

1. Optimization function

Mathematical functions are often used to test the performance of GA. We use average convergence generations (ACGs) to evaluate the performance of GA.

Multimodal mathematical functions are introduced to test the improved GA (IGA) based on the new mutation operator: The 3-D figures of the F functions are illustrated in Fig. 1.

$$F = 0.002 + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^{2} (x_i - a_{ij})^6}$$

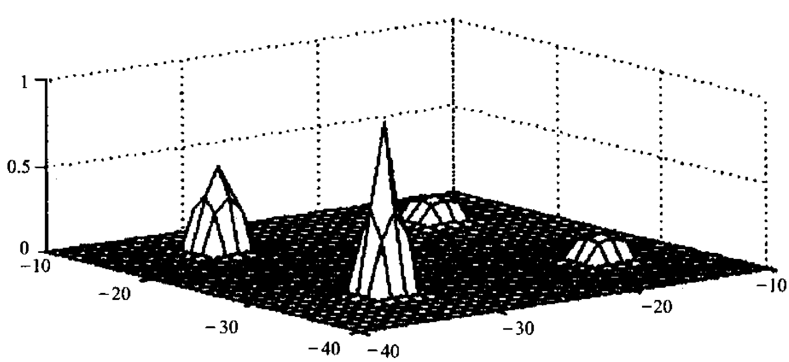

Fig.1 F with $-40 \leqslant (x_1, x_2) \leqslant -10$

2. Influence of Mutation Operator on the Selection of Control Parameters

In this section, we will compare the standard GA (SGA) with IGA.

The population sizes in both SGA and IGA are 20, and the length of encoded strings is 16 for each variable. The probability of crossover P_c ranges from 0.0 − 1.0, and the probability of mutation P_m ranges from 0.0 − 0.3. Fun function is used as the test function. The experiment is repeated 5 times with different initialized population. If SGA and IGA find the value of the function which is larger than 1.0 (called threshold) within 100 generations, the control parameters (P_c, P_m) will be recorded.

Some pairs of (P_c, P_m) meet the above condition. We plot this set of (P_c, P_m) (see "$*$" in the Fig. 2). In SGA, there are only 39 pairs of these kinds of control parameters (P_c, P_m) while the number of (P_c, P_m) increases by 2.5 times in IGA and comes up to 97. That is to say the selection range of control parameters of IGA is wider than that of SGA. The ACGs, which are 53.72 in SGA, decrease to 48.27 in IGA, which also shows that the new mutation operator can accelerate the convergence of GA.

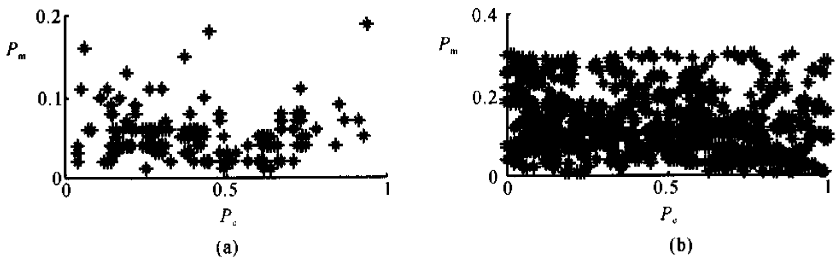

Fig. 2 Range of P_c and P_m in SGA and IGA when optimizing F function
(a) SGA; (b) IGA

3. Comparison with other GAs

Srinivas and Patnaik (1994) presented a GA with adaptive probabilities of crossover and mutation (AGA). In AGA, the mutation operator remains *NOT* operator, but the probabilities of crossover and mutation are determined adaptively for each solution of population.

The size of population is 100 in IGA, and Elitist and tournament competition are adopted. The setting of control parameters for SGA and AGA is seen. The results of comparison are shown in Table 2.

Table 2 Comparison of performance with SGA, AGA, AND IGA

Function	String length	ACGs			Getting Stuck			Threshold	Max number of generation
		SGA	AGA	IGA	SGA	AGA	IGA		
F	17	64.06	36.63	23.23	7	0	0	1.00	100

Obviously, the ACGs in IGA are less than that in SGA and AGA. IGA did not get stuck into the local extreme. Namely the improved GA with the new mutation operator can find the expected solutions when optimizing. SGA and AGA converged prematurely several times in our experiments.

CONCLUSIONS

In this paper, we proved theoretically in this paper that the traditional mutation operator "*NOT*" does not have enough ability to prevent *PCG*; and develop a new mutation operator synthesized by two types of boolean operators. The influence of the new mutation operator on the diversity of genes in the same locus is also discussed. Experiments were conducted using *F* functions to examine the capability to prevent premature convergence. The experimental results revealed that the improved mutation operator enables GA to have more wide selection range of control parameters and to prevent premature convergence more effectively.

References

Back, T., Hoffmeister, F., 1991. Extended selection mechanisms in genetic algorithms, Proc. 4th Int. Conf. Genetic Algorithms, Univ of California, San Diego, CA, p. 14 – 21.

Eshelman, L. J., Shaffer, J. D., 1991. Preventing premature convergence in Genetic algorithms by preventing incest, Proc. 4th Int. Conf. Genetic Algorithms, p. 115 – 122.

Grefenstette, J. J., 1986. Optimization of control parameters for genetic algorithms, *IEEE Trans. Syst., Man, Cybren.*, **16**(1): 122 – 128.

Grefenstette, J. J., Gopal, R. R., and Van Gucht, D., 1985. Genetic Algorithms for the Traveling Salesman Problem, Proc. Int. Conf. Genetic Algorithms and Their Applications, p. 160 – 168.

Holland, J. H., 1975. Adaptation in Natural and Artificial Systems. Ann Arbor, MI: Univ. Michigan Press.

Lin Feng, 2001. New sensorless speed vector control of induction motor drive system, *Journal of Zhejiang University* (Engineering Science), **35**(1): 67 – 71(in Chinese, with English abstrast).

Miller, J. A., Potter, W. D., Gandham R. V., 1993. An evaluation of local improvement operators for genetic algorithms. *IEEE Trans. Syst., Man, Cybern.*, **23** (5): 1340 – 1350.

Potts, J. C., Giddens, T. D. and Yadev, S. B., 1994. The Development and evaluation of an improved genetic algorithm based on migration and artificial selection, *IEEE Trans. Syst., Man, Cybern.*, **24**(1): 73 – 85.